**Project Design Phase-II**

|  |  |
| --- | --- |
| **Date** | **25 June 2025** |
| **Team ID** | **LTVIP2025TMID24654** |
| **Project Name** | **Shopez : one-stop shop for online purchases** |
| **Mentor Name** | **Dr Shaik Salma Begam** |
| **Maximum Marks** | **4 Marks** |

**Technology Stack (Architecture & Stack)**

**Technical Architecture**

Shopez is a MERN stack e-commerce platform designed for scalability, security, and a seamless user experience. The architecture includes a React-based frontend, Node.js/Express backend, MongoDB database, and cloud/file storage for assets. The system is modular, supporting future integration with external APIs and cloud infrastructure.

**Table 1: Technology Stack Components**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology / Service Used |
| 1 | User Interface | Web UI for users and admins | React.js, HTML5, CSS3, JavaScript |
| 2 | Application Logic-1 | User authentication, registration, session management | Node.js, Express.js, JWT |
| 3 | Application Logic-2 | Product browsing, search, filtering, cart management | Node.js, Express.js |
| 4 | Database | Data storage for users, products, orders, banners | MongoDB, Mongoose |
| 5 | Cloud Database | (Optional/Scalable) Cloud-hosted MongoDB | MongoDB Atlas |
| 6 | File Storage | Product images, banners | Local filesystem, (optionally AWS S3) |
| 7 | External API-1 | (Optional) Payment gateway integration | Stripe API, PayPal API |
| 8 | External API-2 | (Optional) Email notifications | SendGrid, Nodemailer |
| 9 | Machine Learning Model | (Future) Product recommendation, personalization | Python (Flask API), TensorFlow, Scikit-learn |
| 10 | Infrastructure | Application deployment, scaling, and management | Local server, Docker, AWS EC2, Heroku |

**Table 2: Application Characteristics**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology / Approach Used |
| 1 | Open-Source Frameworks | Use of open-source frameworks for rapid development and community support | React.js, Node.js, Express.js, MongoDB |
| 2 | Security Implementations | JWT authentication, password hashing, HTTPS, input validation, role-based access control | JWT, bcrypt, Helmet, CORS, OWASP practices |
| 3 | Scalable Architecture | Modular codebase, RESTful APIs, stateless backend, cloud-ready deployment | Docker, AWS, MongoDB Atlas, Microservices (future) |
| 4 | Availability | High uptime via cloud deployment, load balancing, and database replication | AWS EC2, Heroku, MongoDB Atlas, Nginx |
| 5 | Performance | Optimized queries, caching, CDN for static assets, efficient API design | MongoDB indexing, Redis (future), Cloudflare CDN |